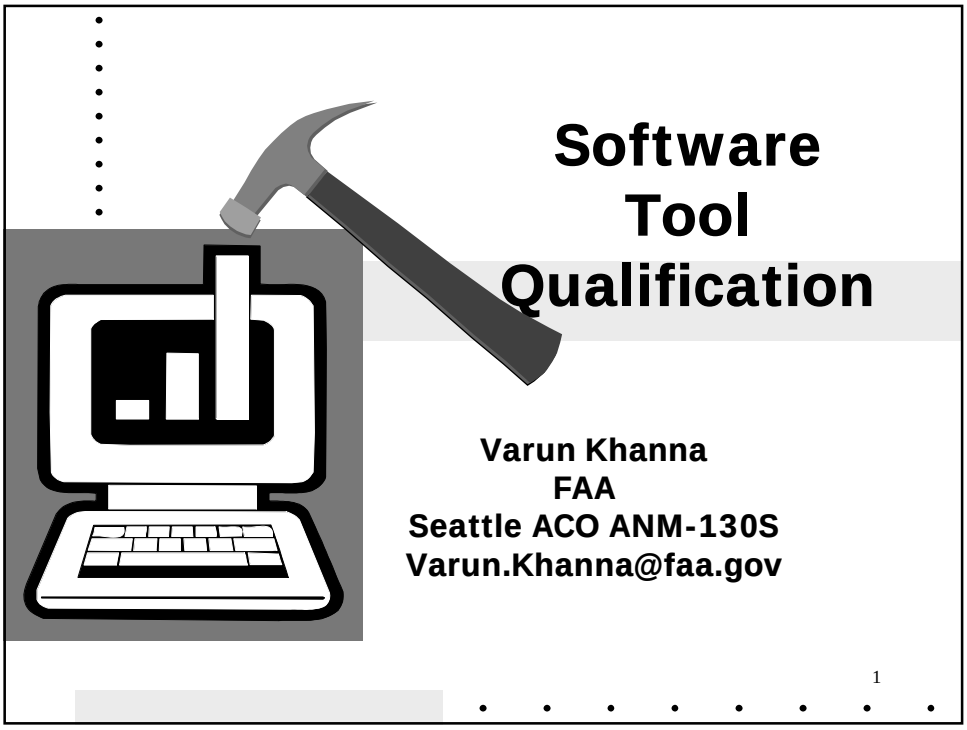
FAA National Software Conference, June 2001
Software Tool Qualification



Relevant References

- RTCA DO-178B/EUROCAE ED-12B
  - Software Life Cycle Environment (4.1c., 4.1d., 4.2c., 4.4, 4.4.1, 4.4.2, 4.4.3, 6.4.1, 11.4b.(9) (control), 11.15 (CI))
  - Error Prevention/Detection (4.2c., 5.1.2b., 5.2.2f., 5.3.2d., 5.4.2c., 6.1, 6.2e., 6.3.1-6.3.4, 6.4)
  - Tool Identification (11.2c., 11.3d., 11.4a., 11.5a., 11.6c., 11.7d., 11.8e., 11.15(SECI), 11.16h.)
  - Tool Qualification (11.1g., 11.20g., 12.2)
- FAA Notice N8110.83, **"Guidelines for the Qualification of Software Tools Using RTCA DO-178B", dated 4/9/99**

2

Purpose

- **Purpose of notice:**
  - Provide Guidelines To ACO Engineers and DERs For Software Tool Qualification
  - Clarify Difference Between Development and Verification Tools
  - Clarify DO-178B Guidance On Tools Qualification

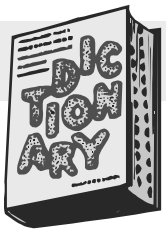
3

History

- **History of Notice:**
  - Identified At Streamlining Software Aspects Of Certification (SSAC) Workshop #1 (Jan 1998) As Confusing Part of DO-178B
  - SSAC Workshop #2 (May 1998) - Began Work On Position
  - Draft Notice Routed For Comments - Sept 98
  - Notice Completed - April 1999

4

What Is A Tool? 1/4

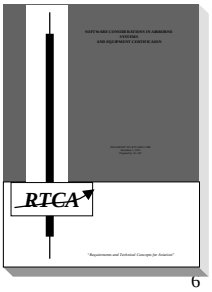


- Dictionary
  - An instrument
  - A Means To An End
  - Anything Used in Performing an Operation
  - Anything Regarded as Necessary to the Carrying Out of One's Occupation or Profession
  - One that is used or manipulated by another

5

What Is A Tool? 2/4

- DO-178B Definition: Software Tool:
  - A computer program used to develop, test, analyze, produce, or modify another program or its documentation.



6

What Is A Tool? 3/4

- **DO-178B Defines Two Software Tools:**

  Software Development Tools:
  "Tools whose output is part of airborne software and thus can introduce errors."

  → Tool that can inject an error into the software.

7

What Is A Tool? 4/4

– **Software Verification Tools:**
"Tools that cannot introduce errors, but may fail to detect them."

→ Tool that may fail to detect an error in the software.

8

Technical Info: Examples

Development	Verification
• autocode generators • compilers • software libraries • operating systems	• simulators • emulators • test tools - coverage analyzers • test case generators

9

Pop Quiz

Classify each of the following software tools:

- CM tool that stores all software life cycle data, including object code and loading procedures.
- Tool used to enforce coding standards
- CRC (Cyclic Redundancy Check)

10

So .. What's Tool Qualification?

- Process To Ensure That A Tool Provides Confidence At Least Equivalent To The Processes That Are Eliminated, Reduced, or Automated
- See DO-178B, Section 12.2
- Alternative: Verification of Tools Outputs per DO-178B Section 6
- Notice N8110.83 Provides Guidelines

11

Notice Outline

- 7 Sections:
  - Section 1: Purpose
  - Section 2: Distribution
  - Section 3: Related Publications
  - Section 4: Background
  - Section 5: Discussion
  - Section 6: Procedures
  - Section 7: Conclusion

12

Background (Section 4) - 1/3

- Tools Are Developed to Eliminate, Reduce, or Automate Portions of the Process
- Obtain Confidence by Qualification
- DO-178B, Section 12.2 Addresses Tool Qualification
- Section 12.2 $\rightarrow\rightarrow$ 8 Areas of Misunderstanding

13

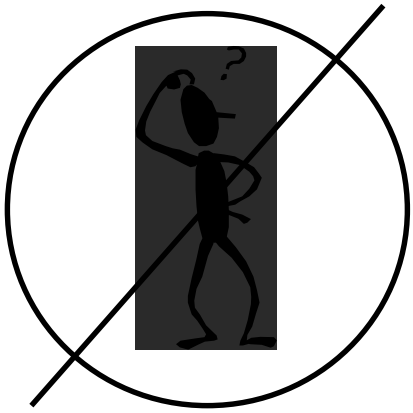
Background (Section 4) - 2/3

8 Areas of Misunder-standing

1. When to Qualify A Tool
2. Different Types of Criteria
3. Applicable Criteria for Tool Qualification
4. Data Production for Tool Qualification
5. Tool Operational Requirements Acceptance Criteria
6. Tool Determinism
7. Tool Partitioning Assurance
8. Tool Configuration Control

14

Background (Section 4) - 3/3

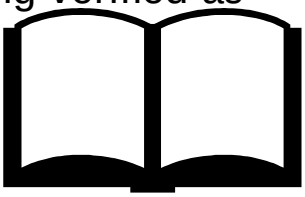


Notice Intended to Clarify

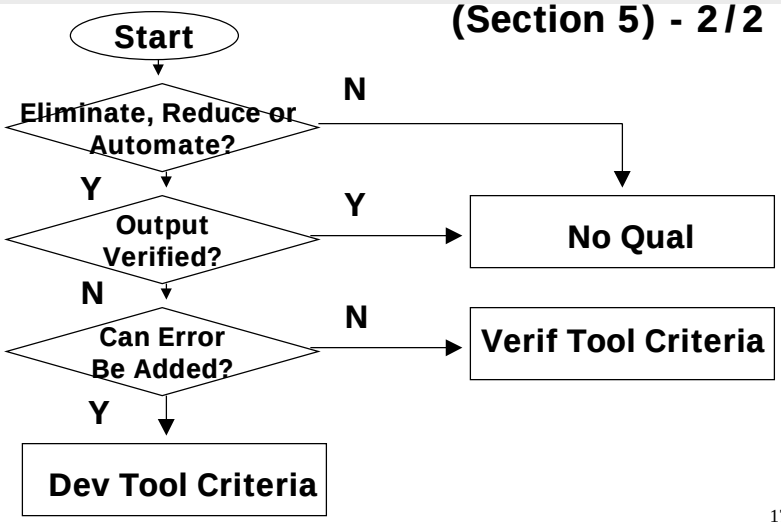
15

When Is Tool Qualification Needed? (Section 5 - Discussion) - 1/2

- DO-178B, 12.2 Tool Qualification states, "Qualification of a tool is needed when processes of this document are eliminated, reduced or automated by the use of a software tool without its output being verified as specified in section 6."



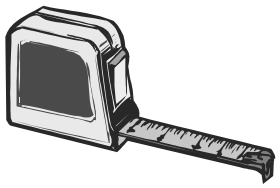
When Is Tool Qualification Needed? (Section 5) - 2/2

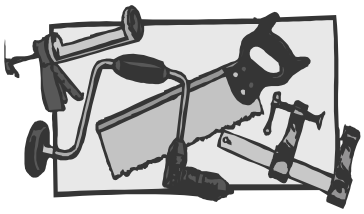


17

Two Types of Tools (Section 5)

1. Verification



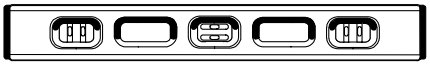


2. Development

18

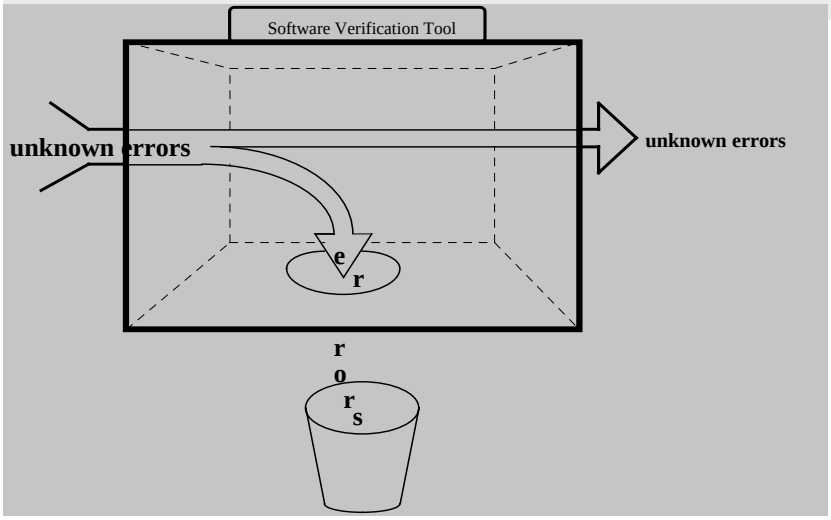
Verification Tools (Section 5) - 1/2

- Tools that cannot introduce errors, but may **fail to detect** them.
  - For example, a static analyzer, that automates a software verification process activity, should be qualified **if the function** that it performs is **not verified** by another activity. Type checkers, analysis tools and test tools are other examples.



19

Verification Tools (Section 5) - 2/2



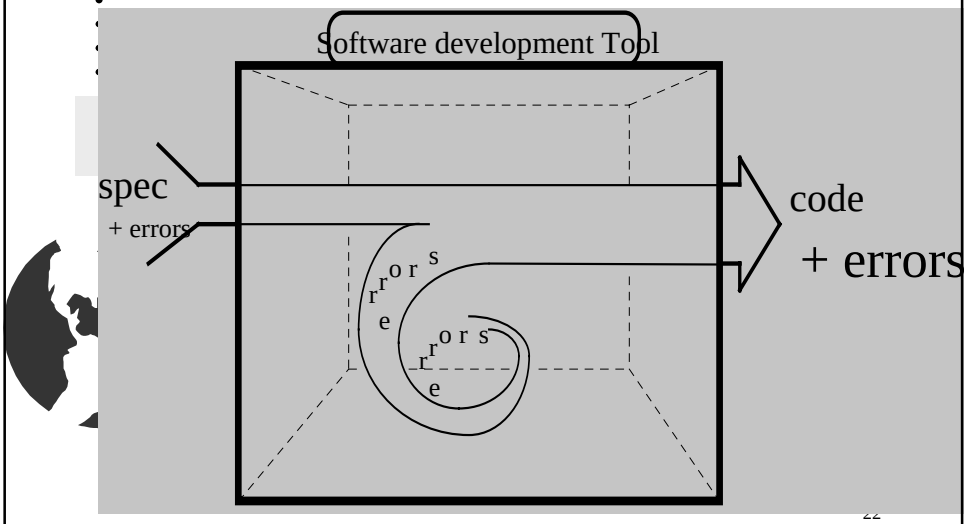
20

Development Tools (Section 5) - 1/2

- Tools whose output is part of airborne software and thus can introduce errors.
- For example, a tool which generates Source Code directly from low-level requirements would have to be qualified if the generated Source Code is not verified as specified in section 6.

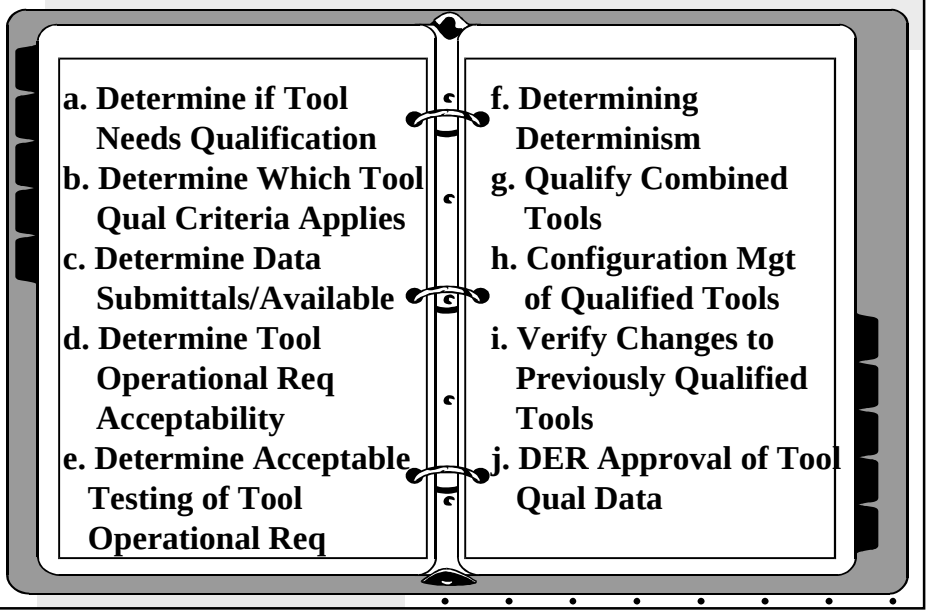
21

Development Tools (Section 5) - 2/2



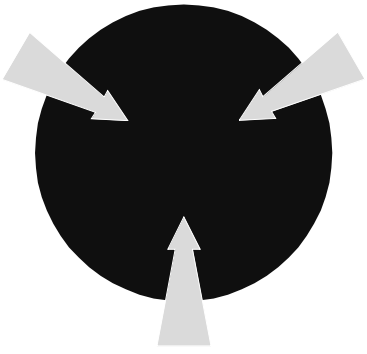
22

10 Guidelines (Section 6)



Three Questions To Determine If Tool Qual Is Needed (Section 6a) - 1/3

1. Can tool insert error or allow an existing error to remain undetected?

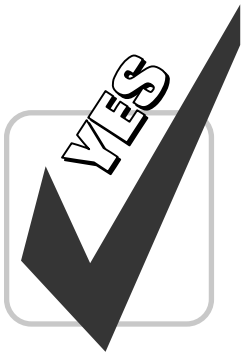


2. Will tool's output not be verified per section 6 of DO-178B?

3. Are processes of DO-178B eliminated, reduced, or automated?

24

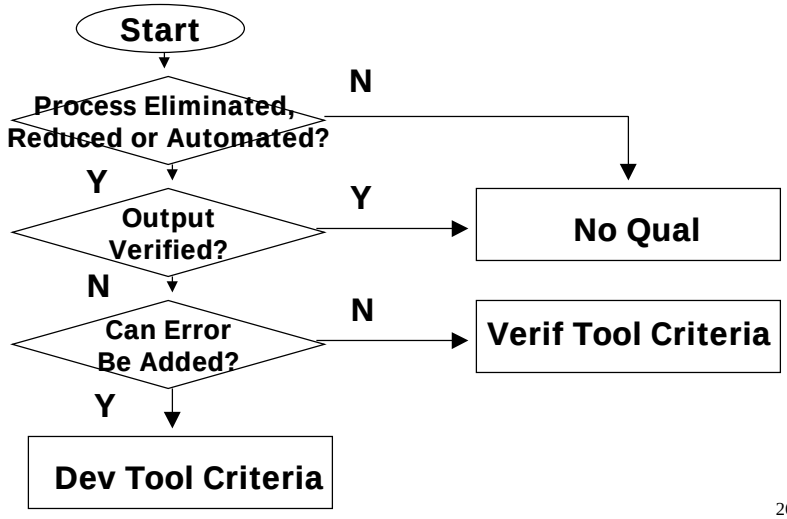
Three Questions To Determine If Tool Qual Is Needed (Section 6a) - 2/3



- **QUALIFY: If answers to 3 questions are YES.**

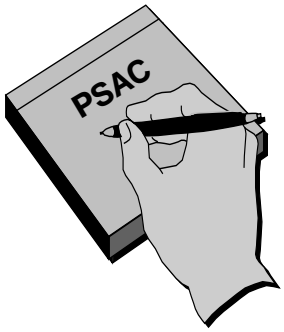
25

Three Questions To Determine If Tool Qual Is Needed (Section 6a) - 3/3



26

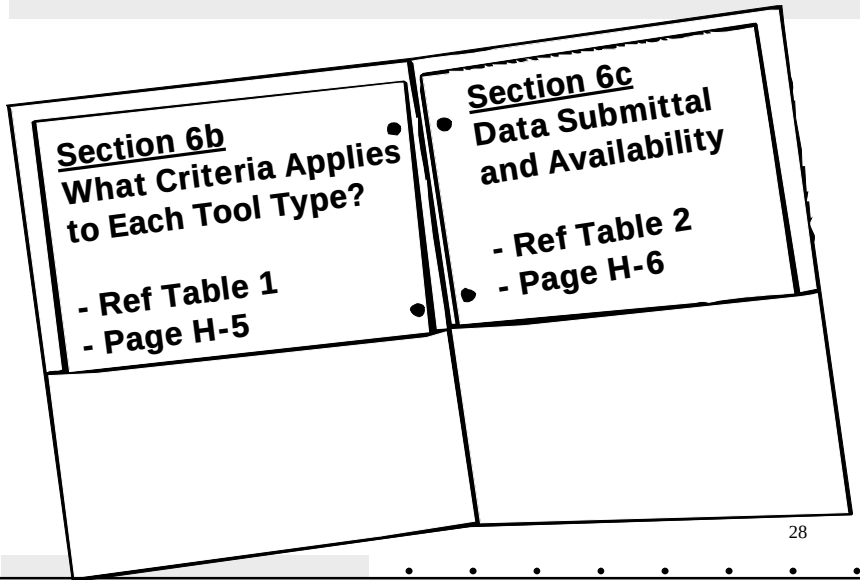
Planning Data (Section 6a)



- All Tools To Be Qualified Should Be Specified in the Plan for Software Aspects of Certification (PSAC, 11.1g.)

27

Sections 6b and 6c



28

Data Submittals (Section 6c)

- **Development tool**
- Plan for Software Aspects of Certification
- Tool Qualification Plan
- Tool Operational Requirements
- Tool Accomplishment Summary
- Software Accomplishment Summary

- **Verification tool**
- Plan for Software Aspects of Certification
- Tool Operational Requirements
- Software Accomplishment Summary

29

Tool Operational Requirements (Section 6d)

- **Development tool**
  - Functionality
  - Operational Environment
  - Installation or Operational Info
  - Development Process Performed
  - Expected Response Under Abnormal Conditions

- **Verification tool**
  - Functionality
  - Operational Environment
  - Installation or Operational Info

30

Tool Operational Requirements
(Section 6e)

- Verification Tools
  - Normal Operating Conditions
  - Only Test Used Portion
- Development Tools
  - Normal Operating Conditions
  - Abnormal Operating Conditions
- See DO-178B Section 6.4.2 For "Normal" vs. "Abnormal"

31

Determinism of Tools
(Section 6f)

- Ability to Establish Correctness of the Output from the Tool
- Given the Same Input, the Tool Should Generate the Correct Output Every Time
  - All Possible Variations of the Output from Some Given Input Should Be Correct
  - Variations in Output Need to be Bounded; e.g., Case/Switch Construct in a Code Generator

32

Combined Tools (Section 6g)

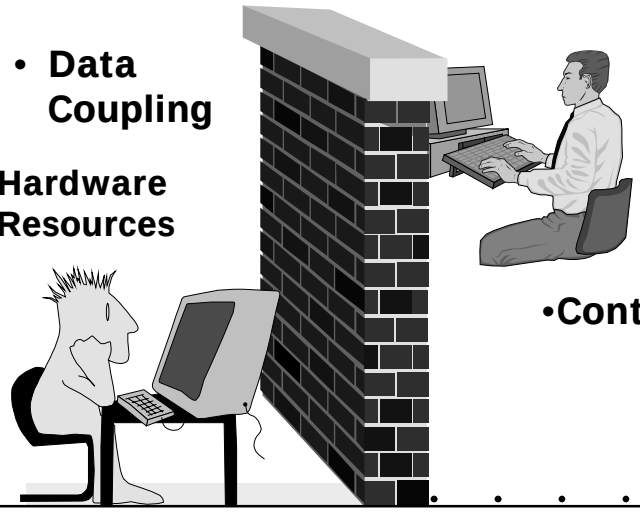
- Output of Both are Used to satisfy a DO-178B Objective
- Tool Functions May Be Qualified Separately IF Partitioning Between Functions Can Be Demonstrated.

33

Examples of Partitioning and Protection (Section 6g)

- Data Coupling

Hardware Resources



- Other Failure Modes
- Control Coupling

34

Configuration Management

(Section 6h)

Configuration Management Control of Qualified Tools:

DO-178B, paragraphs 7.2.9b and 12.2.3b

Tool Qualification Data for Software Development Tools should be controlled as CC1.

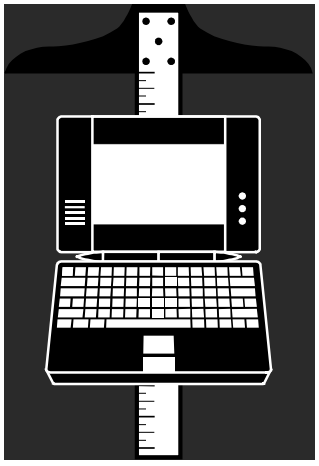
Tool Qualification Data for Software Verification Tools should be controlled as CC2.

35

Section 6i and 6j

6i: Changes to Qualified Tools
Change Impact Analysis

6j: DERs
Don't Delegate if Alternate Means or Policy Issues Exist



36

FAA National Software Conference, June 2001
Software Tool Qualification